```
 s  reset? server

        159323 RESET?
          8694 SERVER
L20          4 RESET? SERVER
                 (RESET?(W)SERVER)

=> d l20 1-


1.  5,857,074, Jan. 5, 1999, Server controller responsive to various
communication protocols for allowing remote communication to a host
computer connected thereto; Derrick W. Johnson, 395/200.47, 200.6 [IMAGE
AVAILABLE]

2.  5,852,720, Dec. 22, 1998, System for storing display data during
first time period prior to failure of computer and during second time
period after reset of the computer; R. Scott Gready, et al., 395/200.47
[IMAGE AVAILABLE]

3.  5,806,065, Sep. 8, 1998, Data system with distributed tree indexes
and method for maintaining the indexes; David B. Lomet, 707/8;
395/200.53; 707/3; 711/130 [IMAGE AVAILABLE]

4.  5,796,566, Aug. 18, 1998, Printed circuit board having conductors
which can be decoupled for isolating inactive integrated circuits
connected thereto; Dinesh Sharma, et al., 361/86; 307/66, 86, 130;
361/92; 365/229 [IMAGE AVAILABLE]
```

```
=> s  reset? server

        159323 RESET?
          8694 SERVER
L20          4 RESET? SERVER
                 (RESET?(W)SERVER)

=> d 120 1-
```

1. 5,857,074, Jan. 5, 1999, Server controller responsive to various communication protocols for allowing remote communication to a host computer connected thereto; Derrick W. Johnson, 395/200.47, 200.6 [IMAGE AVAILABLE]

2. 5,852,720, Dec. 22, 1998, System for storing display data during first time period prior to failure of computer and during second time period after reset of the computer; R. Scott Gready, et al., 395/200.47 [IMAGE AVAILABLE]

3. 5,806,065, Sep. 8, 1998, Data system with distributed tree indexes and method for maintaining the indexes; David B. Lomet, 707/8; 395/200.53; 707/3; 711/130 [IMAGE AVAILABLE]

4. 5,796,566, Aug. 18, 1998, Printed circuit board having conductors which can be decoupled for isolating inactive integrated circuits connected thereto; Dinesh Sharma, et al., 361/86; 307/66, 86, 130; 361/92; 365/229 [IMAGE AVAILABLE]

```
=> d 120 1- hit


US PAT NO:     5,857,074 [IMAGE AVAILABLE]             L20: 1 of 4
```

DETDESC:

DETD(26)

Server controller 26 is configured to perform an automatic server recovery ("ASR") reset of server 10. When the server controller 26 detects a failure of server 10, it waits a predetermined amount of time (typically 30 seconds) and **resets server** 10.

DETDESC:

DETD(27)

The following is an exemplary sequence of a reset, followed by a failure, followed by a reset on server 10. Server 10 is first powered on which causes server 10 to reset. On **reset, server** controller 26 flushes information from previous reset sequence buffer 56, flushes failure sequence buffer 60 and flushes failure start buffer 64. During flushes of buffers 56, 60 and 64, a swap of previous reset sequence buffer 56 and the current reset sequence buffer 54 occurs simultaneous with swaps between buffers 60 and 58 as well as swaps between buffers 64 and 62. Thus, reset causes movement of information from buffers 54, 58 and 62 to buffers 56, 60 and 64, respectively. On **reset, server** 10 software begins performing the POST operations and displaying characters on display 24 Server controller 26 stores the screen changes associated with the POST operations in current reset sequence buffer 54 and current sequence buffer 58. Once the current reset sequence buffer 54 becomes full, the server controller 26 stops storing screen changes thereto. If the current sequence buffer becomes full, the server controller 26 removes the oldest screen changes from the current sequence buffer 58 to make room for the new screen changes and modifies the current start buffer 62 to reflect the old screen changes. Thus, buffer 58 performs similar to a FIFO register, with overflow information sent to buffer 62. As described below, a combination of buffers 58 and 62 (when swapped with bufers 60 and 64) depict the entire failure sequence of video screens prior to the most recent failure.

DETDESC:

DETD(28)

When a failure of server 10 occurs, in accordance with the second step in the reset-failure-reset three-step example, a sequence of video screens occur upon display 24 indicative of that failure and possibly the cause of that failure. These screens are automatically displayed as part of many server operating systems available as ABEND messages from Novell Netware or a Microsoft NT blue screen. Thirty seconds after failure occurs, server controller 26 performs an ASR reset on server 10. Thus, the second reset within the reset-failure-reset example occurs during ASR reset. The reset, similar to the first reset, enables server controller 26 to flush buffers 56, 60 and 64 while swapping buffers 56, 60 and 64 with buffers 54, 58 and 62, respectively. Similar to the first **reset**, **server** 10 again initiates POST operations and forwards characters to display 24. Server controller 26 stores the screen changes occurring after reset, resulting from POST operation, in current reset sequence buffer 54 and current sequence buffer 58. Previous reset sequence buffer 56 now contains the sequence of screen changes associated with the initial power on reset, and the failure sequence buffer 60 as well as failure start buffer 64 now contain the sequence of screen changes leading up to that failure. As opposed to the initial reset contained within the previous reset sequence buffer 56, current reset sequence buffer 54 contains the reset associated with the second reset (or ASR reset).

US PAT NO: 5,852,720 [IMAGE AVAILABLE] L20: 2 of 4

DETDESC:

DETD(26)

Server controller 26 is configured to perform an automatic server recovery ("ASR") reset of server 10. When the server controller 26 detects a failure of server 10, it waits a predetermined amount of time (typically 30 seconds) and **resets server** 10.

DETDESC:

DETD(27)

The following is an exemplary sequence of a reset, followed by a failure, followed by a reset on server 10. Server 10 is first powered on which causes server 10 to reset. On **reset, server** controller 26 flushes information from previous reset sequence buffer 56, flushes failure sequence buffer 60 and flushes failure start buffer 64. During flushes of buffers 56, 60 and 64, a swap of previous reset sequence buffer 56 and the current reset sequence buffer 54 occurs simultaneous with swaps between buffers 60 and 58 as well as swaps between buffers 64 and 62. Thus, reset causes movement of information from buffers 54, 58 and 62 to buffers 56, 60 and 64, respectively. On **reset, server** 10 software begins performing the POST operations and displaying characters on display 24. Server controller 26 stores the screen changes associated with the POST operations in current reset sequence buffer 54 and current sequence buffer 58. Once the current reset sequence buffer 54 becomes full, the server controller 26 stops storing screen changes thereto. If the current sequence buffer becomes full, the server controller 26 removes the oldest screen changes from the current sequence buffer 58 to make room for the new screen changes and modifies the current start buffer 62 to reflect the old screen changes. Thus, buffer 58 performs similar to a FIFO register, with overflow information sent to buffer 62. As described below, a combination of buffers 58 and 62 (when swapped with bufers 60 and 64) depict the entire failure sequence of video screens prior to the most recent failure.

DETDESC:

DETD(28)

When a failure of server 10 occurs, in accordance with the second step in the reset-failure-reset three-step example, a sequence of video screens occur upon display 24 indicative of that failure and possibly the cause of that failure. These screens are automatically displayed as part of many server operating systems available as ABEND messages from Novell Netware or a Microsoft NT blue screen. Thirty seconds after failure occurs, server controller 26 performs an ASR reset on server 10. Thus, the second reset within the reset-failure-reset example occurs during ASR reset. The reset, similar to the first reset, enables server controller 26 to flush buffers 56, 60 and 64 while swapping buffers 56, 60 and 64 with buffers 54, 58 and 62, respectively. Similar to the first **reset**, **server** 10 again initiates POST operations and forwards characters to display 24. Server controller 26 stores the screen changes occurring after reset, resulting from POST operation, in current reset sequence buffer 54 and current sequence buffer 58. Previous reset sequence buffer 56 now contains the sequence of screen changes associated with the initial power on reset, and the failure sequence buffer 60 as well as failure start buffer 64 now contain the sequence of screen changes leading up to that failure. As opposed to the initial reset contained within the previous reset sequence buffer 56, current reset sequence buffer 54 contains the reset associated with the second reset (or ASR reset).

US PAT NO: 5,806,065 [IMAGE AVAILABLE] L20: 3 of 4

DETDESC:

DETD(59)

When all bits in the bit vector have been **reset, server** site C tells server site O that the tombstone for the deleted page can be garbage collected. This does not require a two site atomic action. Rather, server site O simply drops the tombstone in a local atomic action. It then notifies server site C that the tombstone is gone. Should either message be lost, server site C will eventually ask server site O to delete the tombstone again and server site O will comply or simply report that this has been done already.

US PAT NO: 5,796,566 [IMAGE AVAILABLE] L20: 4 of 4

DETDESC:

DETD(26)

Server controller 26 is configured to perform an automatic server recovery ("ASR") reset of server 10. When the server controller 26 detects a failure of server 10, it waits a predetermined amount of time (typically 30 seconds) and **resets server** 10.

DETDESC:

DETD(27)

The following is an exemplary sequence of a reset, followed by a failure, followed by a reset on server 10. Server 10 is first powered on which causes server 10 to reset. On **reset, server** controller 26 flushes information from previous reset sequence buffer 56, flushes failure sequence buffer 60 and flushes failure start buffer 64. During flushes of buffers 56, 60 and 64, a swap of previous reset sequence buffer 56 and the current reset sequence buffer 54 occurs simultaneous with swaps between buffers 60 and 58 as well as swaps between buffers 64 and 62. Thus, reset causes movement of information from buffers 54, 58 and 62 to buffers 56, 60 and 64, respectively. On **reset, server** 10 software begins performing the POST operations and displaying characters on display 24. Server controller 26 stores the screen changes associated with the POST operations in current reset sequence buffer 54 and current sequence buffer 58. Once the current reset sequence buffer 54 becomes full, the server controller 26 stops storing screen changes thereto. If the current sequence buffer becomes full, the server controller 26 removes the oldest screen changes from the current sequence buffer 58 to make room for the new screen changes and modifies the current start buffer 62 to reflect the old screen changes. Thus, buffer 58 performs similar to a FIFO register, with overflow information sent to buffer 62. As described below, a combination of buffers 58 and 62 (when swapped with bufers 60 and 64) depict the entire failure sequence of video screens prior to the most recent failure.

DETDESC:

DETD(28)

When a failure of server 10 occurs, in accordance with the second step in the reset-failure-reset three-step example, a sequence of video screens occur upon display 24 indicative of that failure and possibly the cause of that failure. These screens are automatically displayed as part of many server operating systems available as ABEND messages from Novell Netware or a Microsoft NT blue screen. Thirty seconds after failure occurs, server controller 26 performs an ASR reset on server 10. Thus, the second reset within the reset-failure-reset example occurs during ASR reset. The reset, similar to the first reset, enables server controller 26 to flush buffers 56, 60 and 64 while swapping buffers 56, 60 and 64 with buffers 54, 58 and 62, respectively. Similar to the first **reset, server** 10 again initiates POST operations and forwards characters to display 24. Server controller 26 stores the screen changes occurring after reset, resulting from POST operation, in current reset sequence buffer 54 and current sequence buffer 58. Previous reset sequence buffer 56 now contains the sequence of screen changes associated with the initial power on reset, and the failure sequence buffer 60 as well as failure start buffer 64 now contain the sequence of screen changes leading up to that failure. As opposed to the initial reset contained within the previous reset sequence buffer 56, current reset sequence buffer 54 contains the reset associated with the second reset (or ASR reset).

```
L1           66 S RESET?(4A)SERVER
L2        11366 S MODEM AND INTERFACE
L3           11 S L1 AND L2
L4         3025 S CLIENT AND SERVER
L5            7 S L3 AND L4

=> d l5 1-


1.  5,857,074, Jan. 5, 1999, Server controller responsive to various
communication protocols for allowing remote communication to a host
computer connected thereto; Derrick W. Johnson, 395/200.47, 200.6 [IMAGE
AVAILABLE]

2.  5,852,720, Dec. 22, 1998, System for storing display data during
first time period prior to failure of computer and during second time
period after reset of the computer; R. Scott Gready, et al., 395/200.47
[IMAGE AVAILABLE]

3.  5,809,311, Sep. 15, 1998, System and method for providing centralized
backup power in a computer system; Craig Steven Jones, 395/750.01;
361/90; 364/528.32, 707; 395/284 [IMAGE AVAILABLE]

4.  5,796,566, Aug. 18, 1998, Printed circuit board having conductors
which can be decoupled for isolating inactive integrated circuits
connected thereto; Dinesh Sharma, et al., 361/86; 307/66, 86, 130;
361/92; 365/229 [IMAGE AVAILABLE]

5.  5,764,899, Jun. 9, 1998, Method and apparatus for communicating an
optimized reply; Gene Eggleston, et al., 395/200.33; 348/6, 7 [IMAGE
AVAILABLE]

6.  5,638,448, Jun. 10, 1997, Network with secure communications
sessions; Minhtam C. Nguyen, 380/29, 9, 21, 23, 25, 37, 43, 49 [IMAGE
AVAILABLE]

7.  5,351,276, Sep. 27, 1994, Digital/audio interactive communication
network; William J. Doll, Jr., et al., 379/88.17; 370/354; 379/68, 88.13,
93.01, 101.01, 247, 269; 707/512 [IMAGE AVAILABLE]
```